Name:

ID:

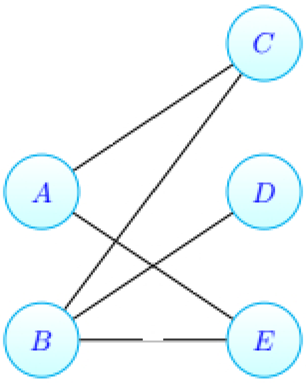
CSCI 3104, Algorithms
Quiz 8 Q1 S17

Profs. Chen & Grochow
Spring 2020, CU-Boulder

Instructions: This quiz is open book and open note. You **may** post clarification questions to Piazza, with the understanding that you may not receive an answer in time and posting does count towards your time limit (30 min for 1x, 37.5 min for 1.5x, 45 min for 2x). Questions posted to Piazza **must be posted as PRIVATE QUESTIONS.** Other use of the internet, including searching for answers or posting to sites like Chegg, is strictly prohibited. Violations of these are grounds to receive a 0 on this quiz. Proofs should be written in **complete sentences. Show and justify all work to receive full credit.**

Standard 17. Let $G(V, E)$ be a graph. A *matching* of G is a set of edge $\mathcal{M}$ such that no two edges in $\mathcal{M}$ share a common vertex. That is, if $(i, j), (u, v) \in \mathcal{M}$ are distinct edges, then $i \neq u, i \neq v$, $j \neq u$, and $j \neq v$.

A graph is *bipartite* if its vertices can be partitioned into two sets $V(G) = L \cup R$ such that every edge has one endpoint in L and one endpoint in R. Note that L and R are disjoint. The graph pictured below is an example.



Consider the following problem

Bipartite Maximum Matching
Input: A bipartite graph $G = (L, R; E)$
Output: A matching $\mathcal{M} \subseteq E(G)$ whose size $|\mathcal{M}|$ is as large as possible.

a. Describe how to reduce the above problem to the (one-source, one-sink) max-flow problem from class. Your description should be **general**, and not tied to a specific example. (You may illustrate with an example for expository purposes, but an example alone is not sufficient. E.g., "This is how my construction is performed in general. Then for example, this is how we apply the construction to the graph I selected.")

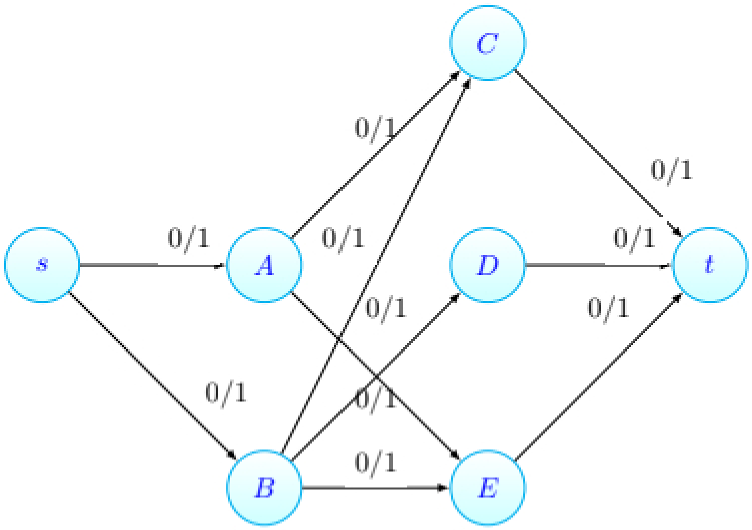
Solution: We construct flow network $\mathcal{N}$ from G as follows.

- $V(\mathcal{N}) = L \cup R \cup \{s, t\}$. That is, the vertices of $\mathcal{N}$ are the vertices of G, and we also add a source s and sink t.
- For each undirected edge $\{i, j\} \in E(G)$, with $i \in L$ and $j \in R$, we add a directed edge (i, j) (that is, $i \to j$) in $E(\mathcal{N})$.
- For each vertex $v \in L$, there is a directed edge (s, v) in $\mathcal{E}(\mathcal{N})$.
- For each vertex $v \in R$, there is a directed edge (v, t) in $E(\mathcal{N})$.
- Each edge in $E(\mathcal{N})$ has capacity 1.

In particular, a maximum flow configuration on $\mathcal{N}$ corresponds precisely to a maximum matching on G by taking the directed edges from L to R that have flow 1.

b. Using your reduction, find a maximum matching in the graph above. Show your work, as well as your final answer. Note that there may be multiple maximum matchings in the graph above; you need only find one such matching.

Solution: We transform the graph on the first page to a flow network as follows.



We now perform the Ford–Fulkerson Algorithm on this flow network. There are six possible ways to perform Ford–Fulkerson.

- We use the Flow Augmenting Paths

$$s \to A \to C \to t$$
$$s \to B \to D \to t.$$

  We push 1 unit of flow along each flow augmenting path. This yields the maximum matching $\{(A, C), (B, D)\}$.

- We use the Flow Augmenting Paths

$$s \to A \to C \to t$$
$$s \to B \to E \to t.$$

  We push 1 unit of flow along each flow augmenting path. This yields the maximum matching $\{(A, C), (B, E)\}$.